Yves Baumann, Tal Ben-Nun, Maciej Besta, Lukas Gianinazzi, Torsten Hoefler, and Piotr Luczynski

Low-Depth Spatial Tree Algorithms

IPDPS 2024

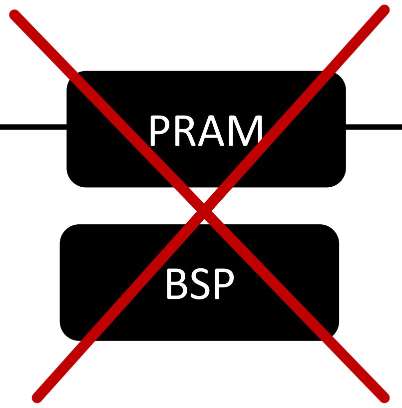
Spatial Dataflow Architectures

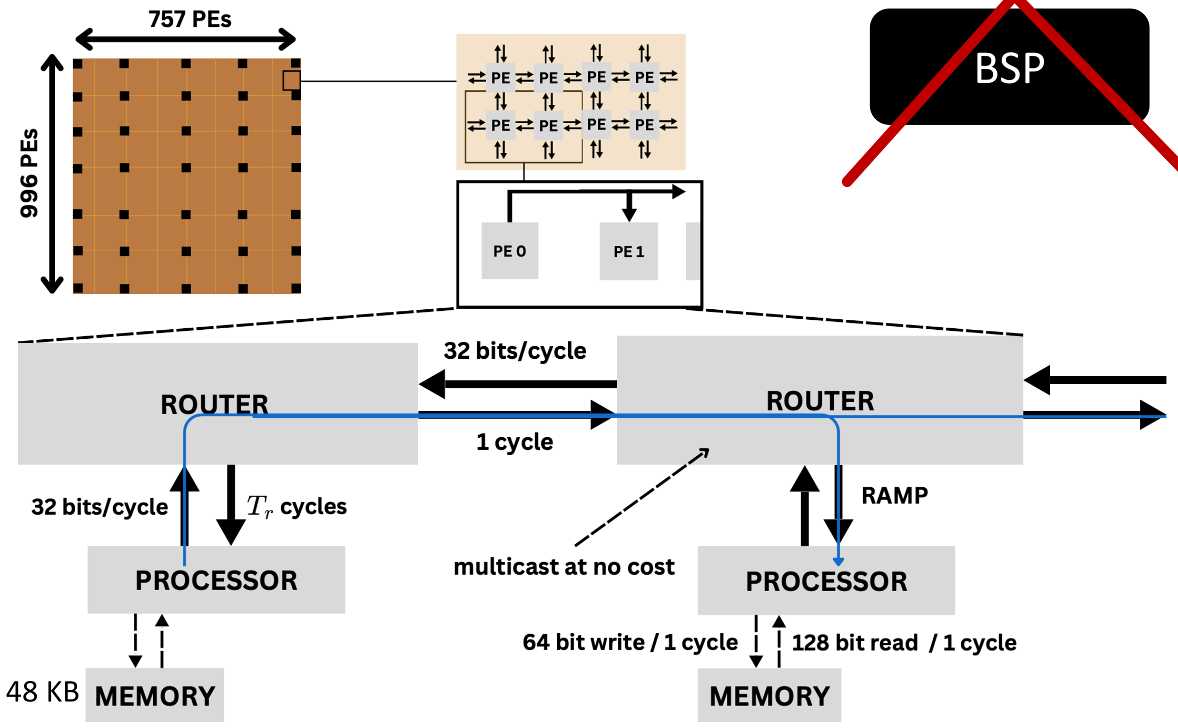
AMD
Versal Adaptive SoCs

Samba Nova
Reconfigurable Dataflow Architecture

...

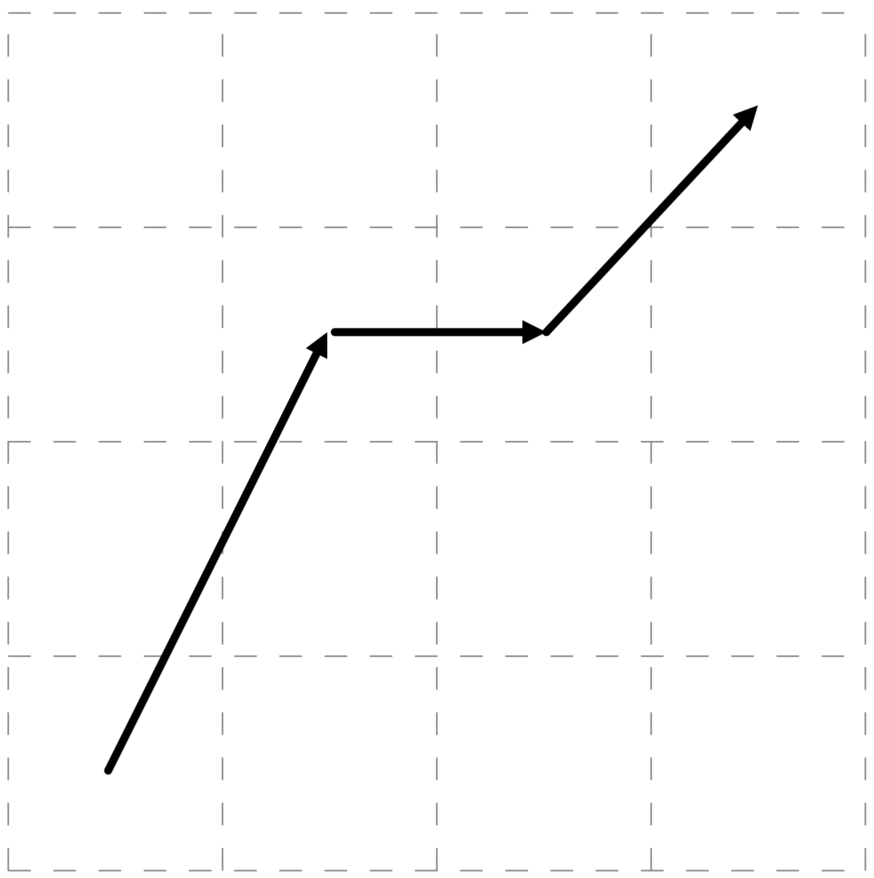
Graphcore
IPUs



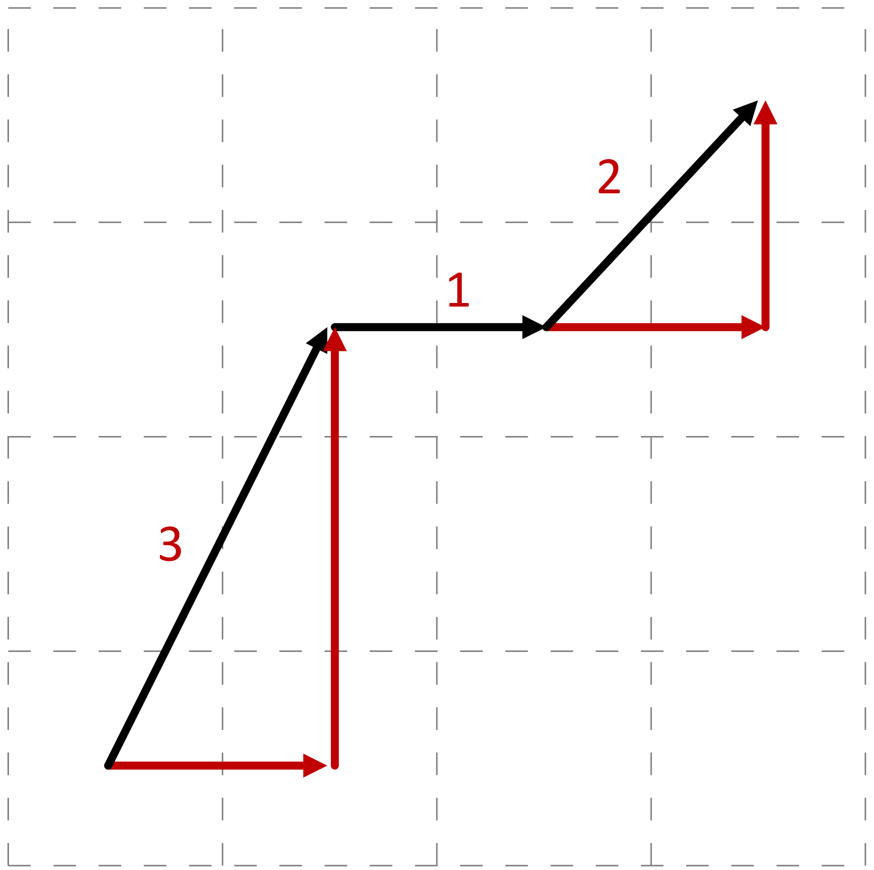


Cerebras
Wafer-Scale Engine

Model of Computation

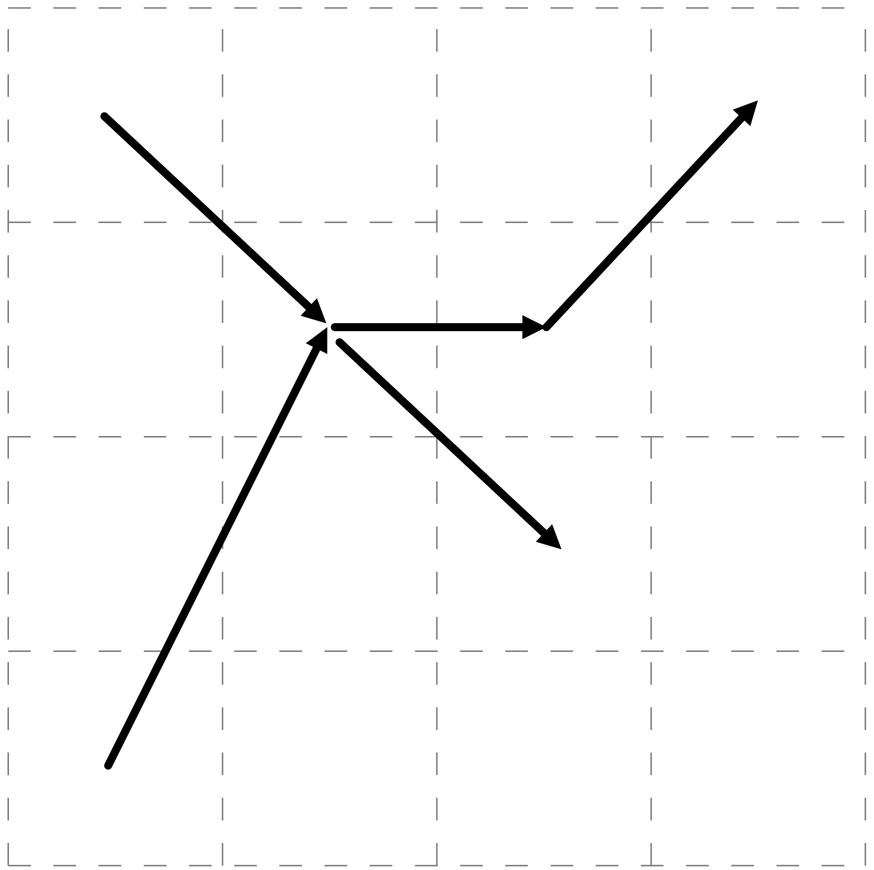


Model of Computation



Distance 6

Model of Computation

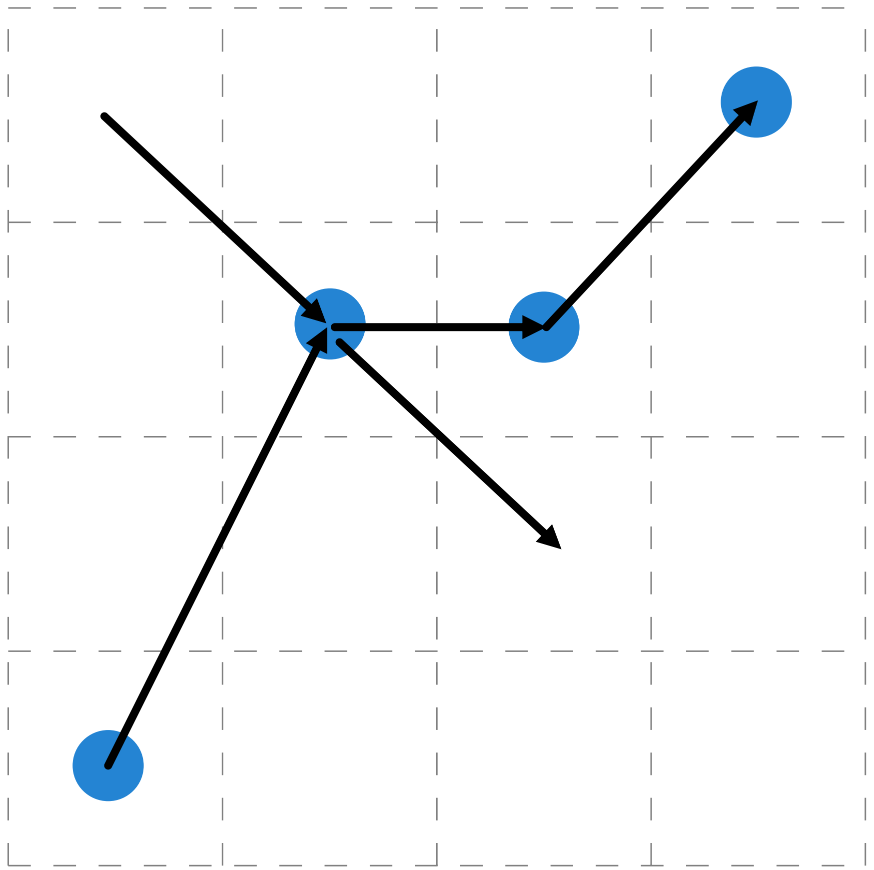


Distance

Maximum 6

Total 10

Model of Computation



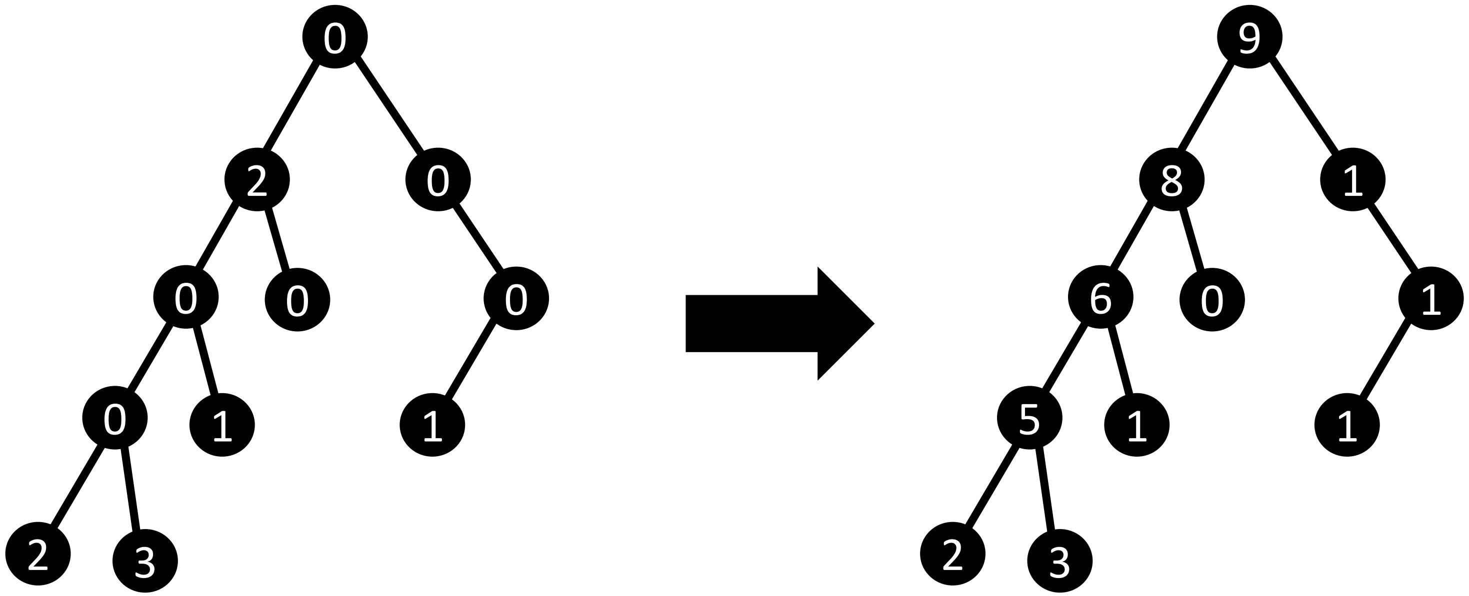
Distance

Maximum 6

Total 10

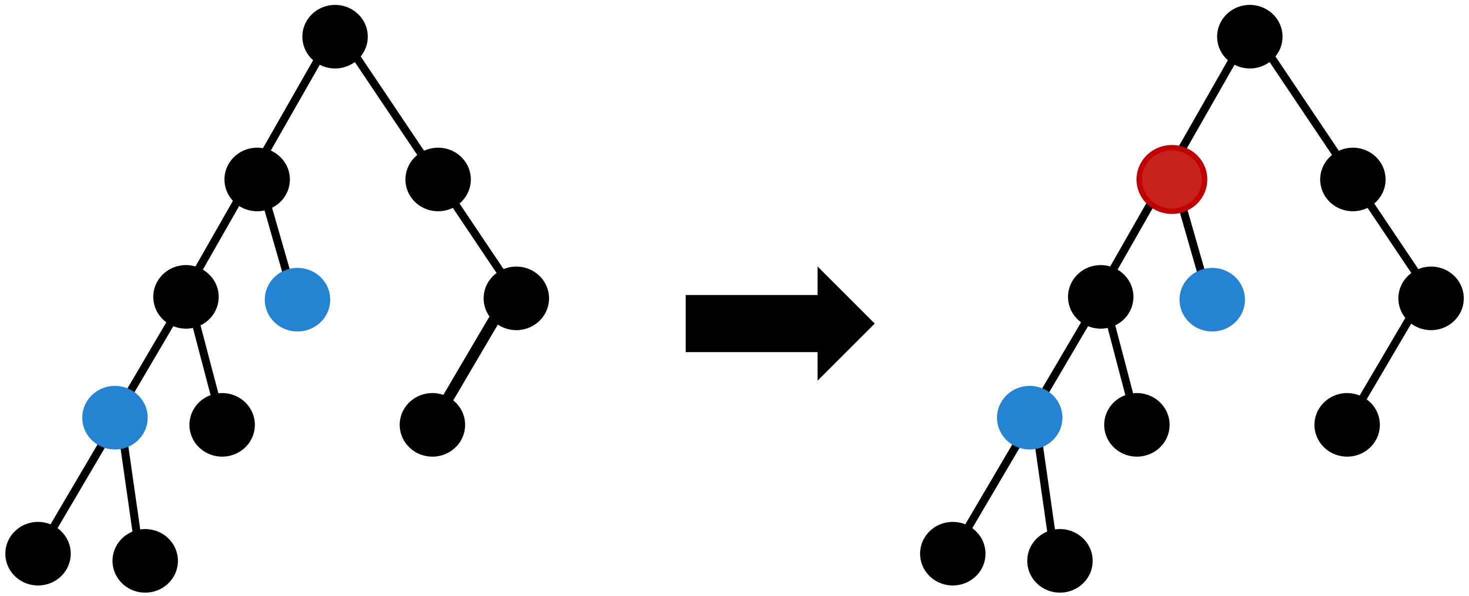
Depth 4

Problems on Trees



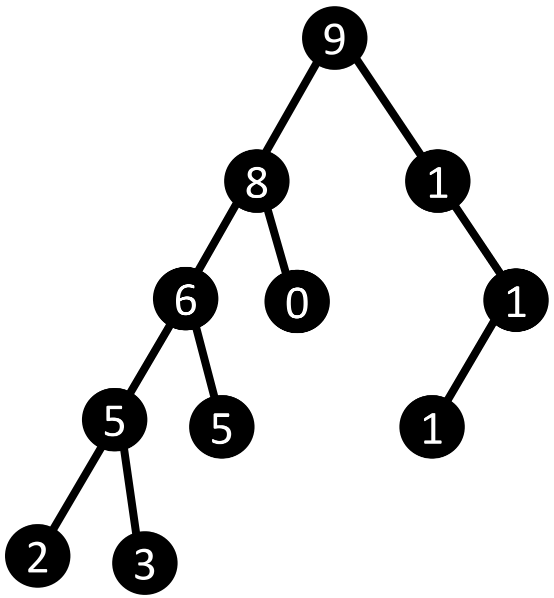
Treefix-Sum

Problems on Trees



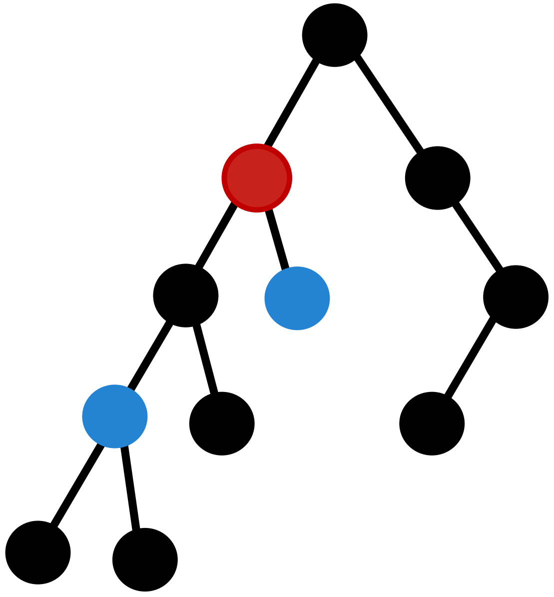
Lowest Common Ancestor (LCA)

Results



Treefix Sum
n vertices

$O(n \log n)$ total distance
$O(\log^2 n)$ depth



LCA
n vertices

$O(n \log^2 n)$ total distance
$O(\log^2 n)$ depth

Challenges

Sparse, Irregular Structures

Costly Random Access

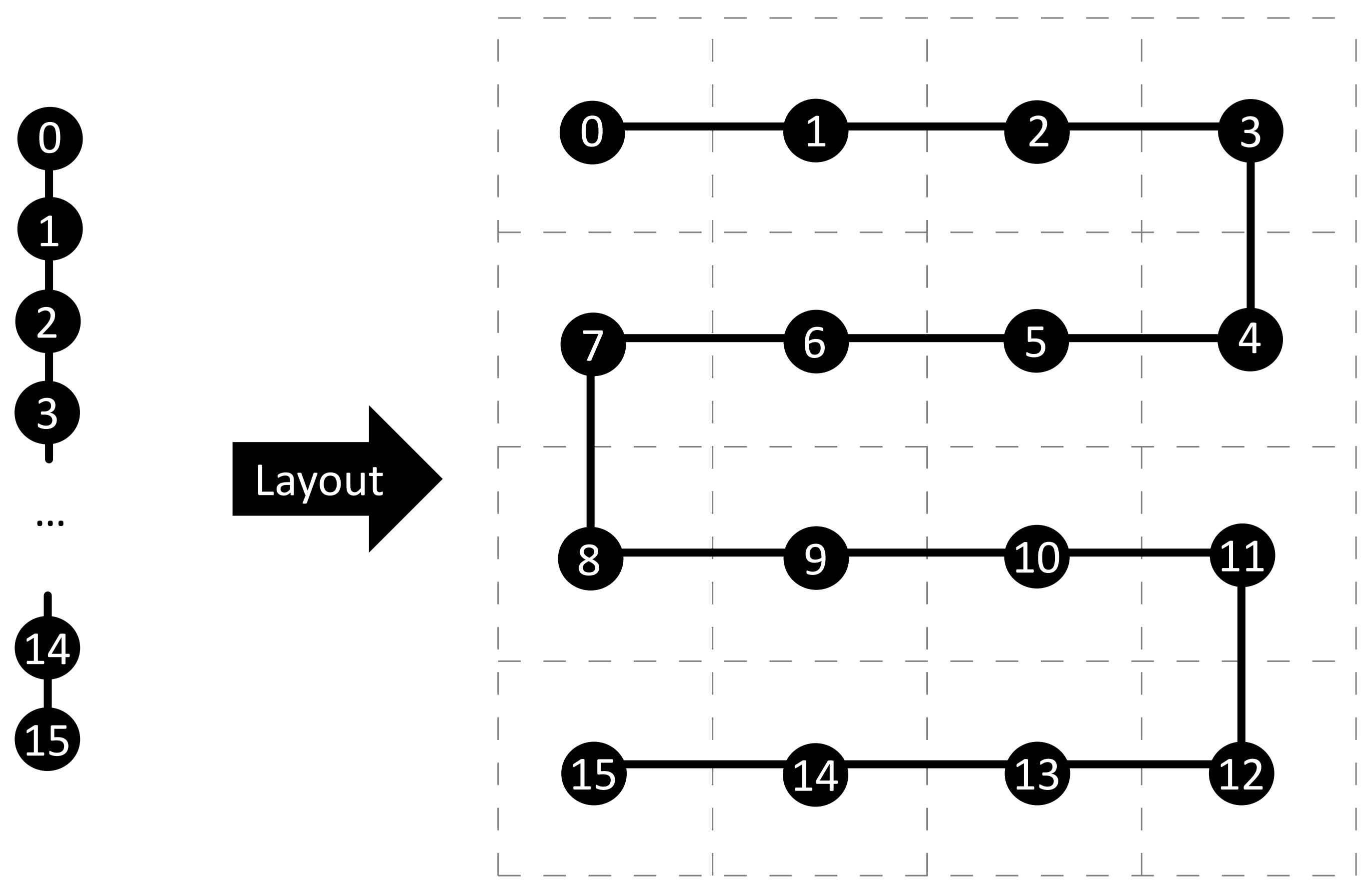
PRAM Simulation:
$\Omega(n\sqrt{n}\ \log n)$ total distance

Our Approach

A. Spatial Layout

B. Logical Operations

Spatial Layout – Paths

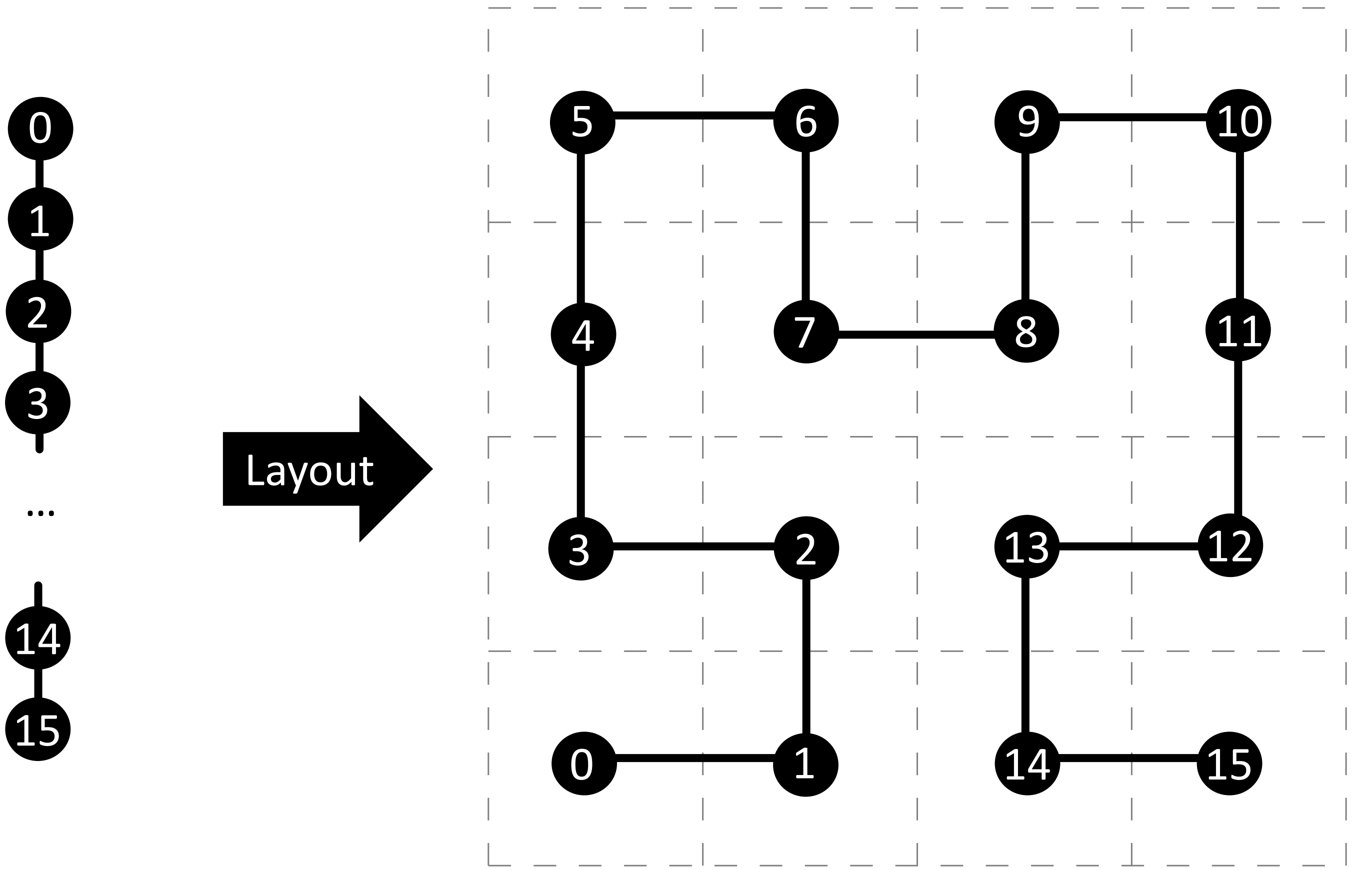


Message a neighbor in the path:

1 hop: distance 1

k hops: distance $\leq \boldsymbol{k}$
average distance $\Theta(\boldsymbol{k})$

Spatial Layout – Paths

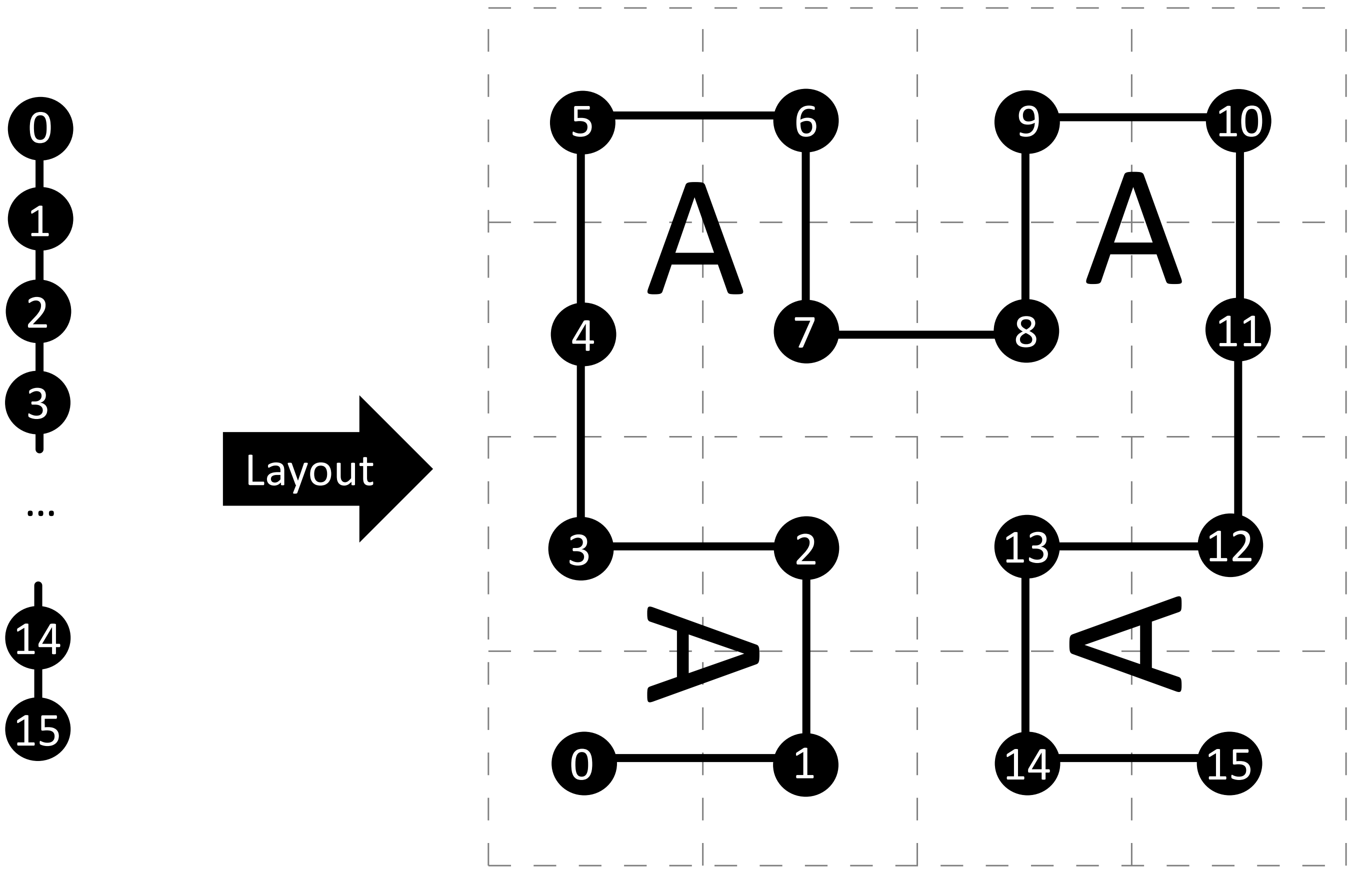


Hilbert Curve (2D)

Message a neighbor in the path:

1 hop: distance 1

Spatial Layout – Paths

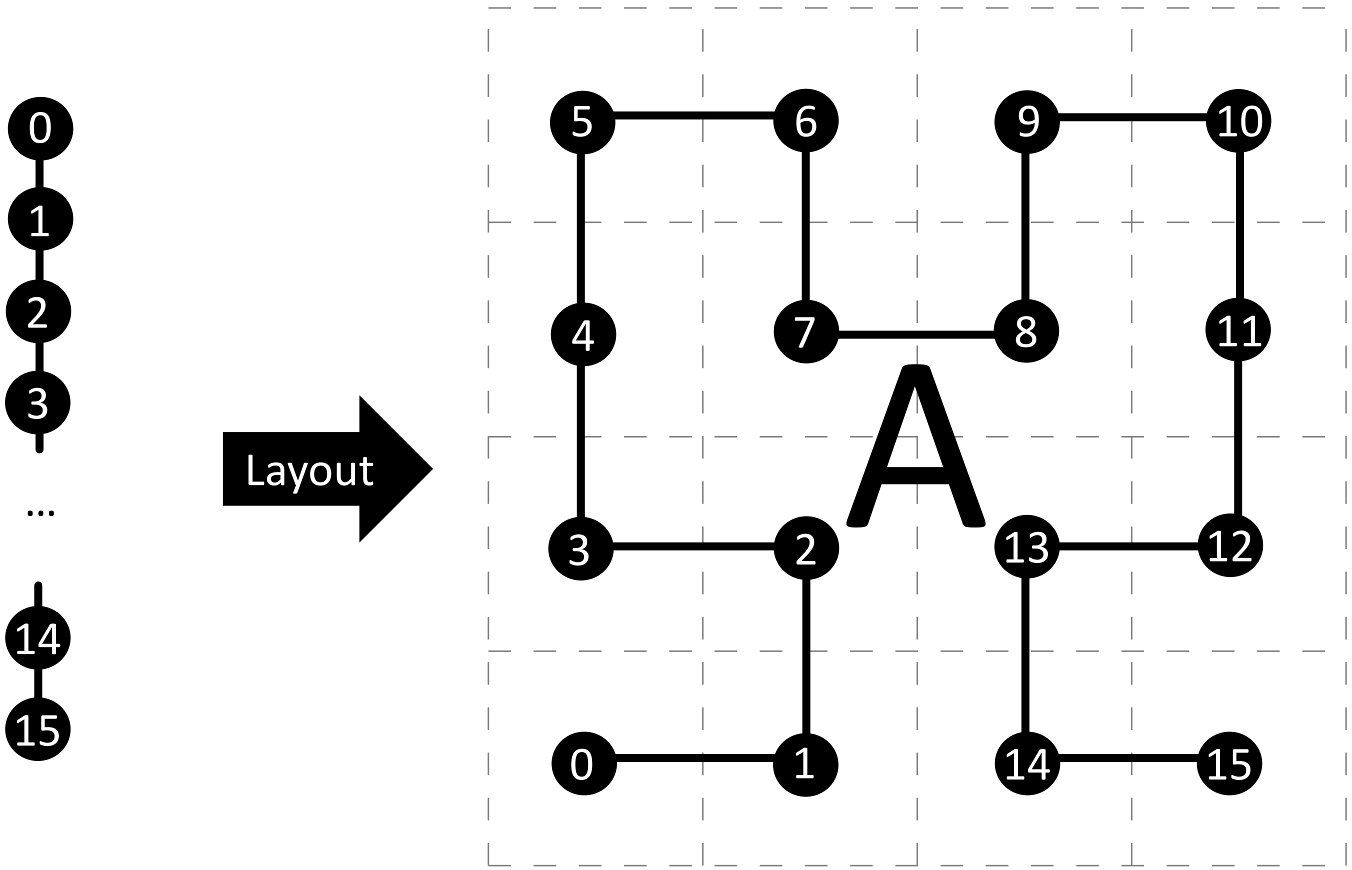


Hilbert Curve (2D)

Message a neighbor in the path:

1 hop: distance 1

Spatial Layout – Paths

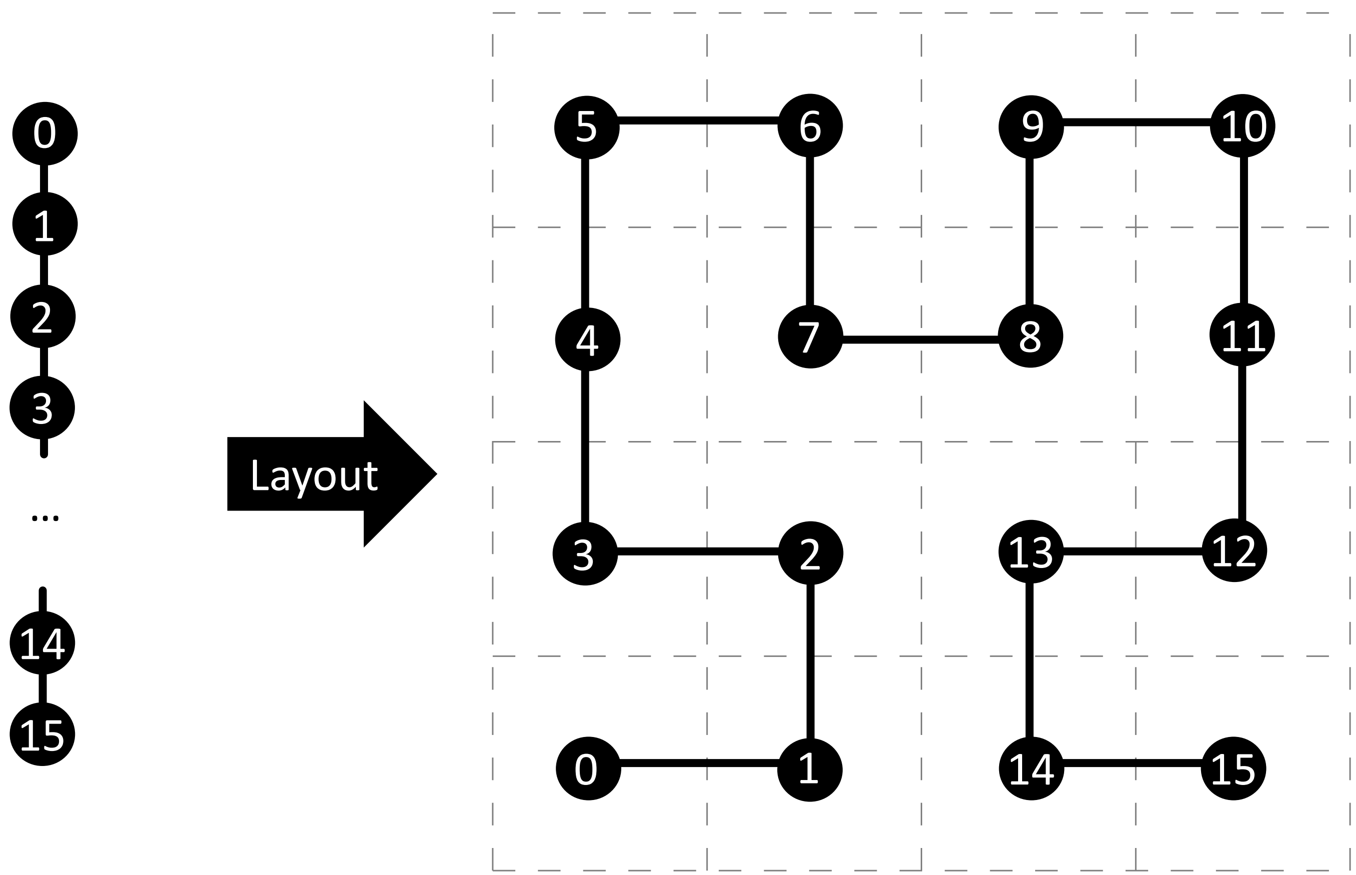


Hilbert Curve (2D)

Message a neighbor in the path:

1 hop: distance 1

Spatial Layout – Paths



Hilbert Curve (2D)

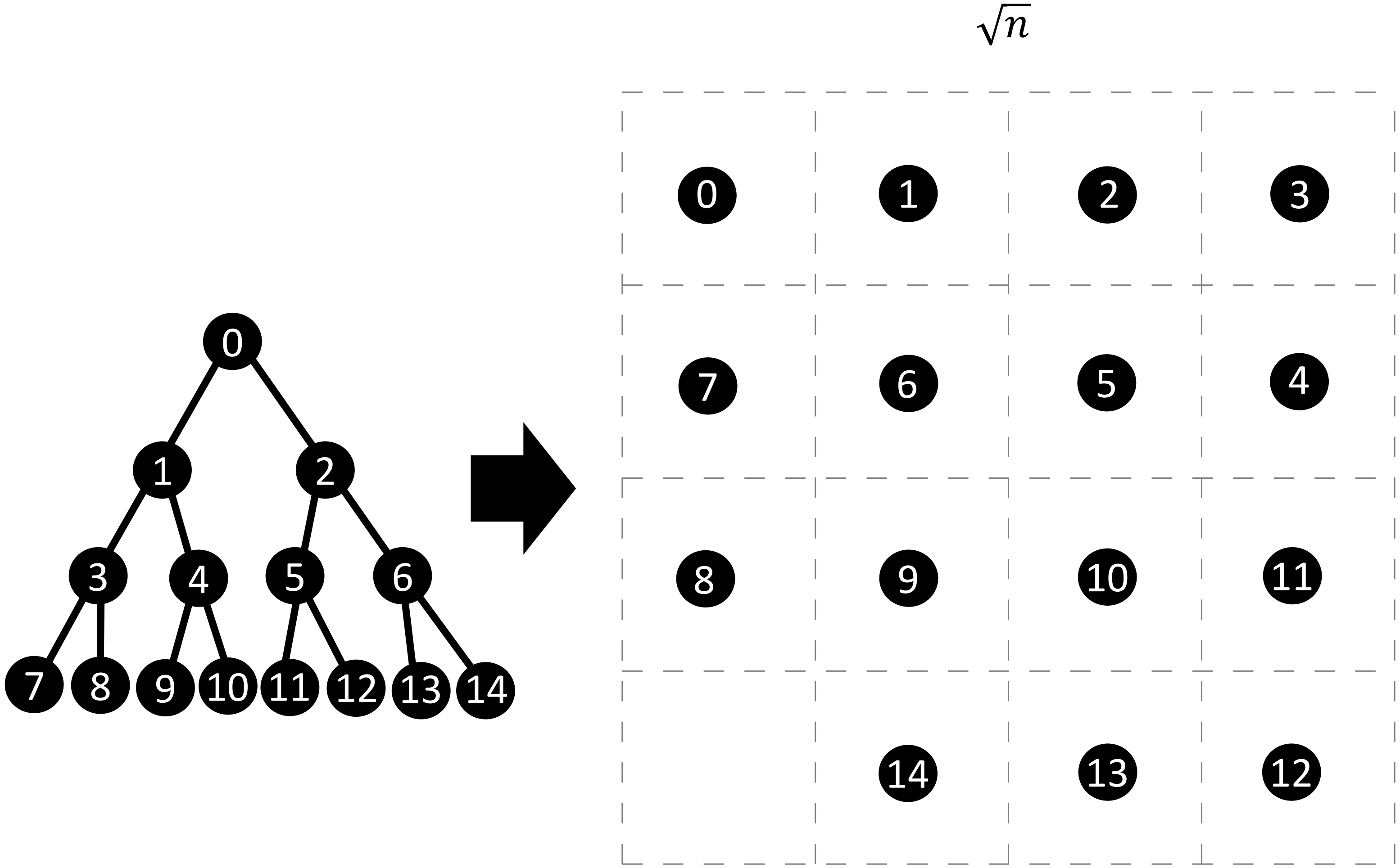
Message a neighbor in the path:

1 hop: distance 1

k hops: distance $\leq 3\sqrt{\boldsymbol{k}} - 2$

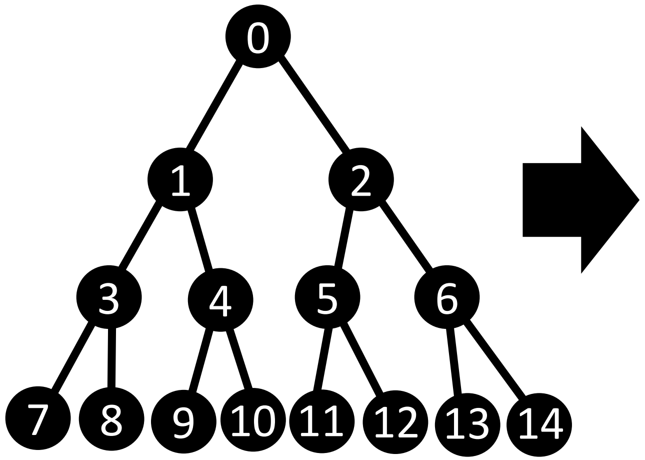
average distance $\Theta(\sqrt{\boldsymbol{k}})$

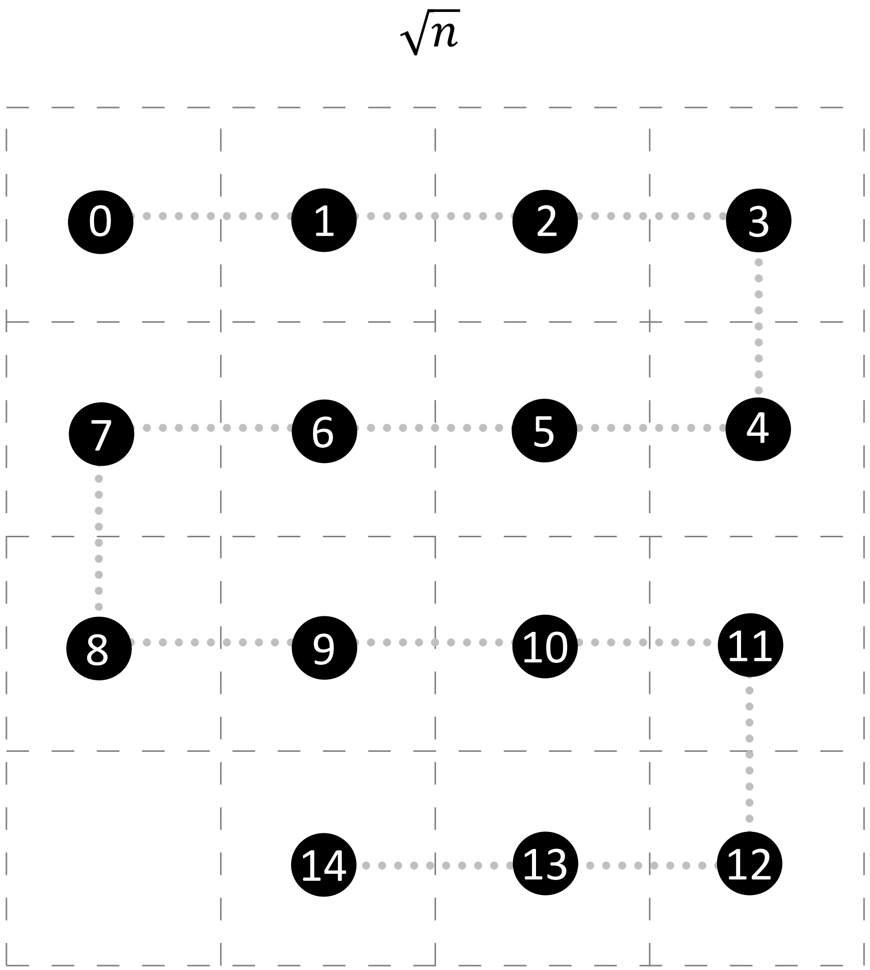
Spatial Layout – Trees



Message a neighbor in the ***tree****:*

Spatial Layout – Trees



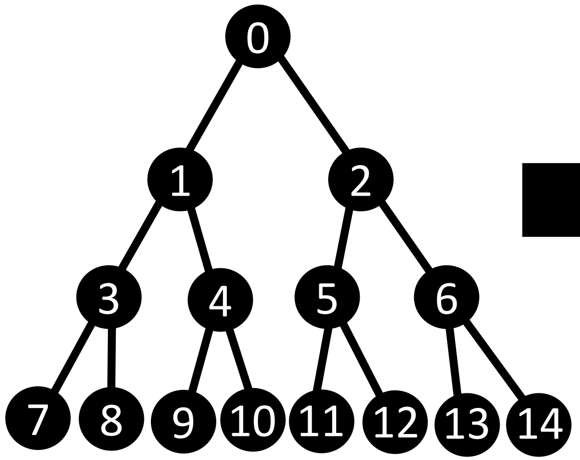


*Message a neighbor in the **tree**:*

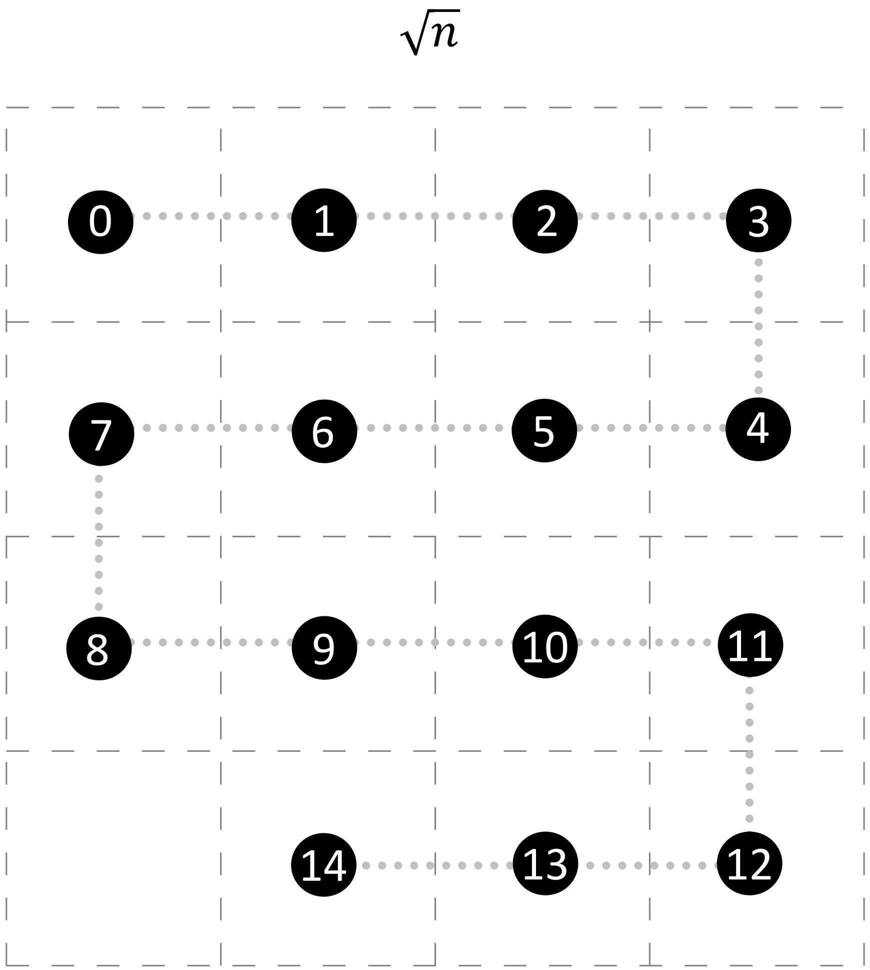
1. **Linearize** Tree

2. Layout According to **2D Curve**

Spatial Layout – Trees



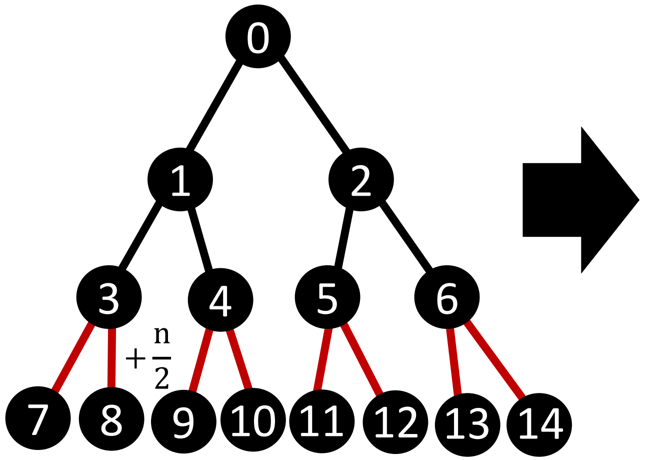
BFS



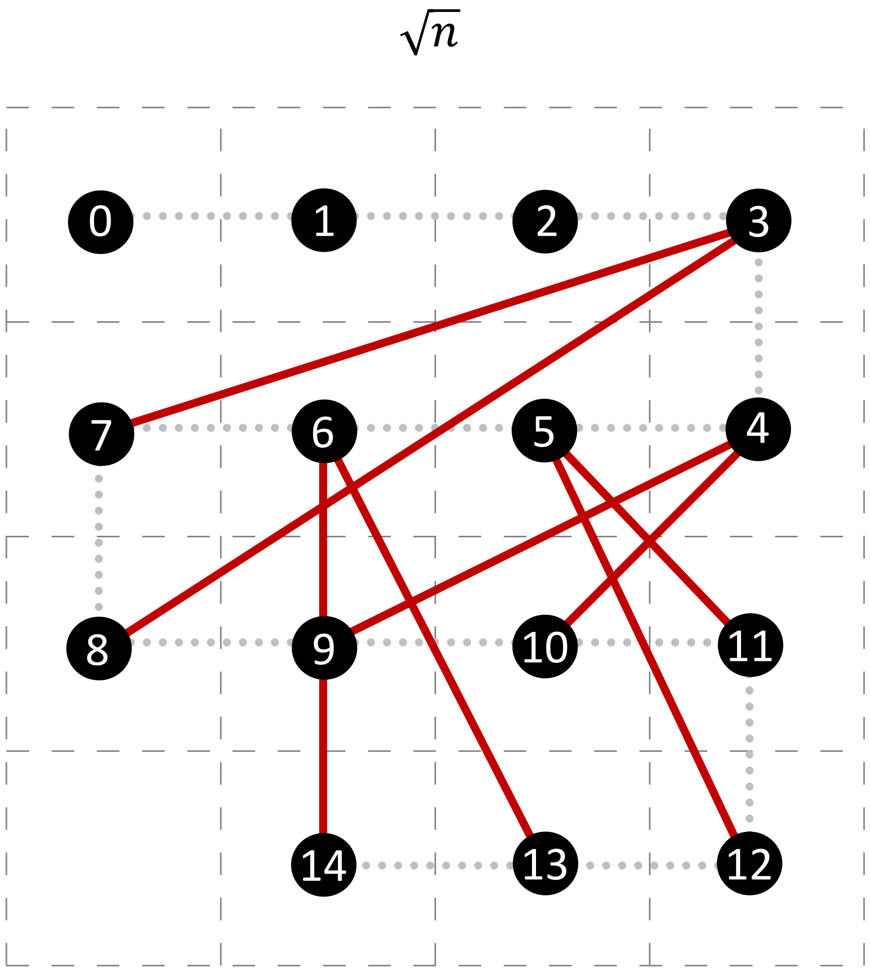
BFS+Snake

*Message a neighbor in the **tree**:*

Spatial Layout – Trees



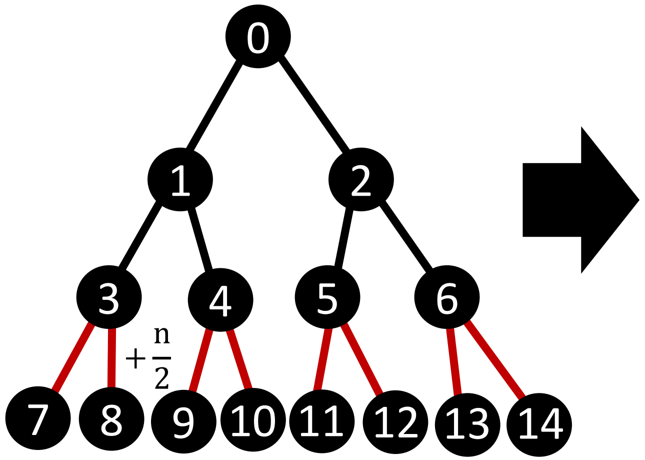
BFS



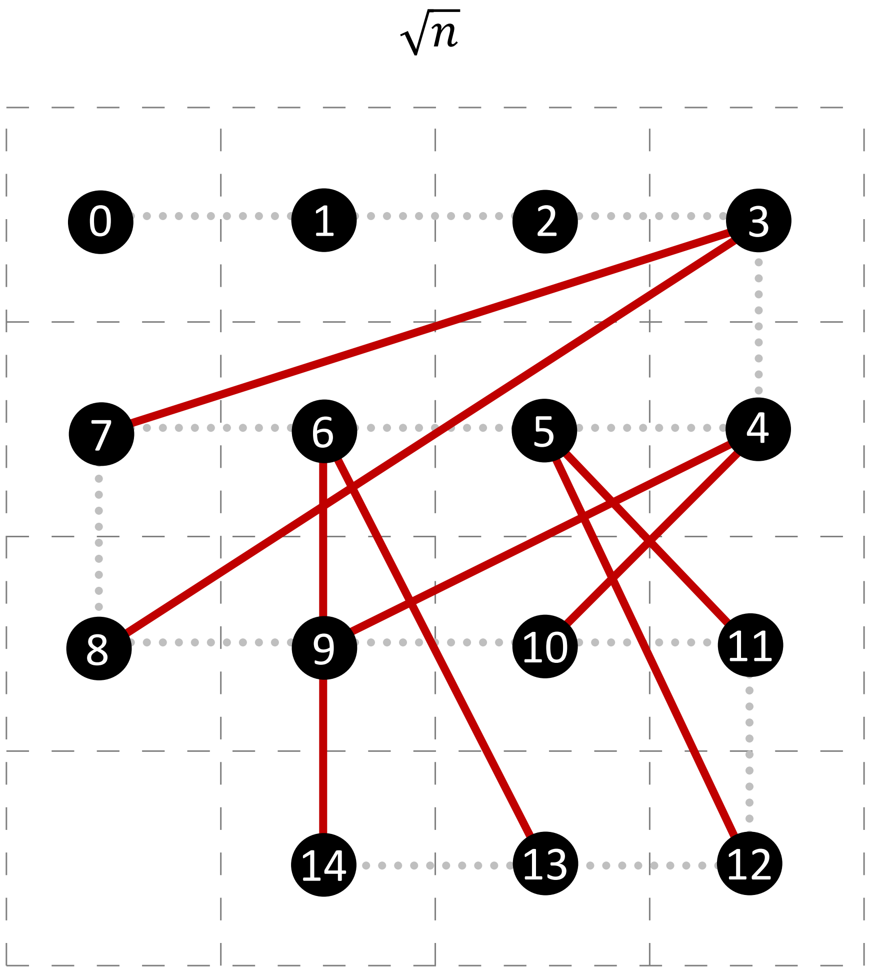
BFS+Snake

Message a neighbor in the ***tree****:*

Spatial Layout – Trees



BFS

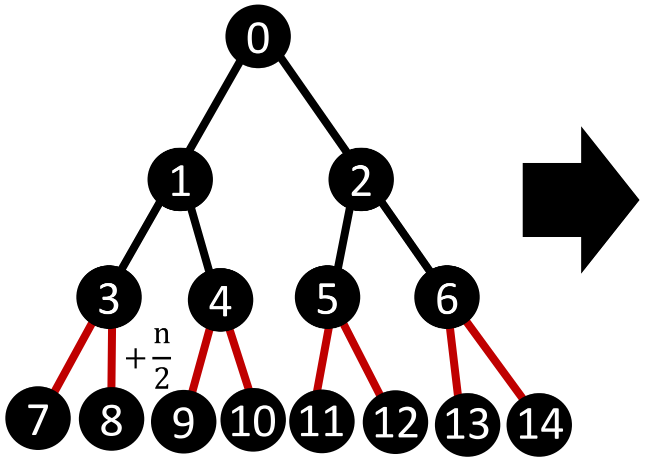


BFS+Snake

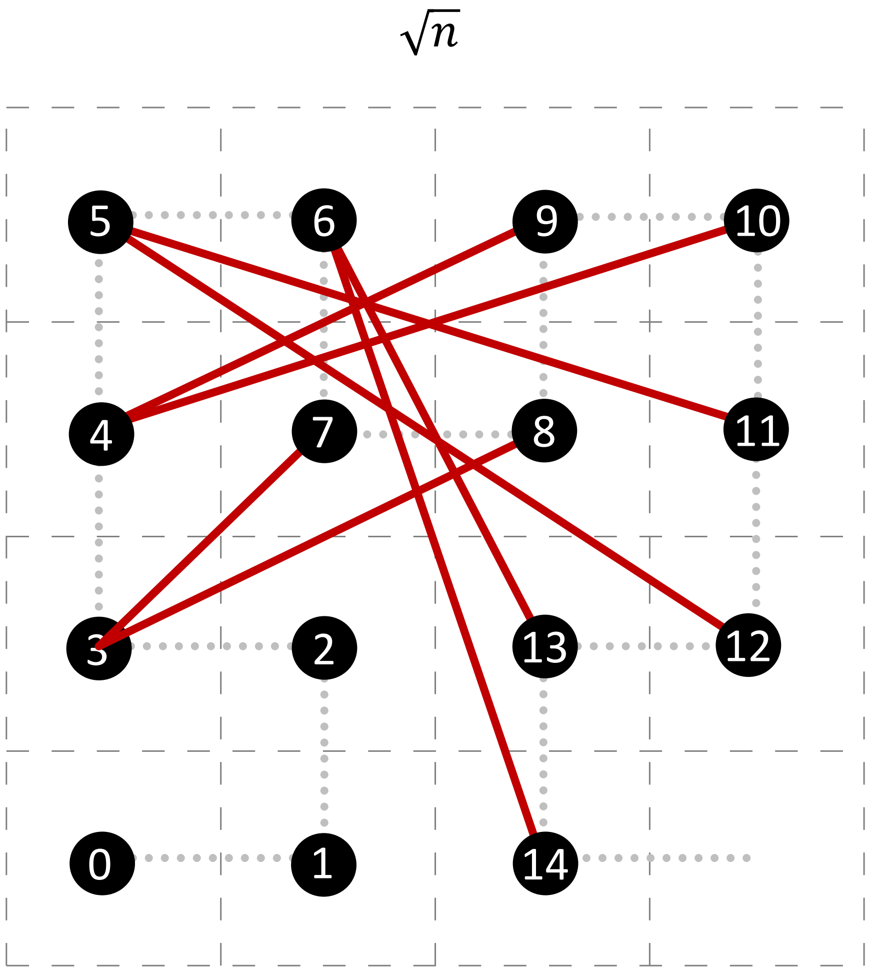
Message a neighbor in the ***tree****:*

1 hop: average distance $\Theta(\sqrt{\boldsymbol{n}})$

Spatial Layout – Trees



BFS

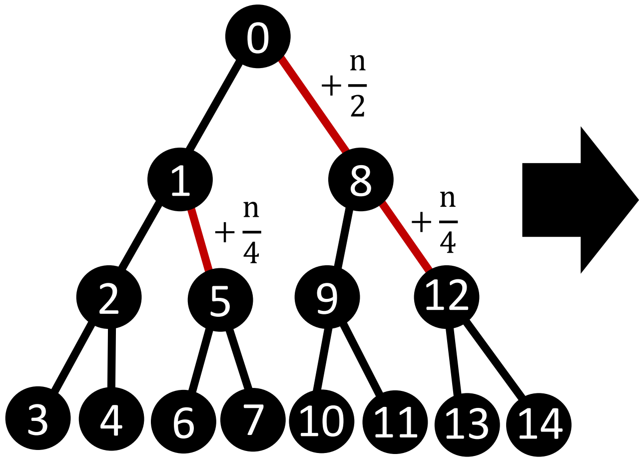


BFS+Hilbert

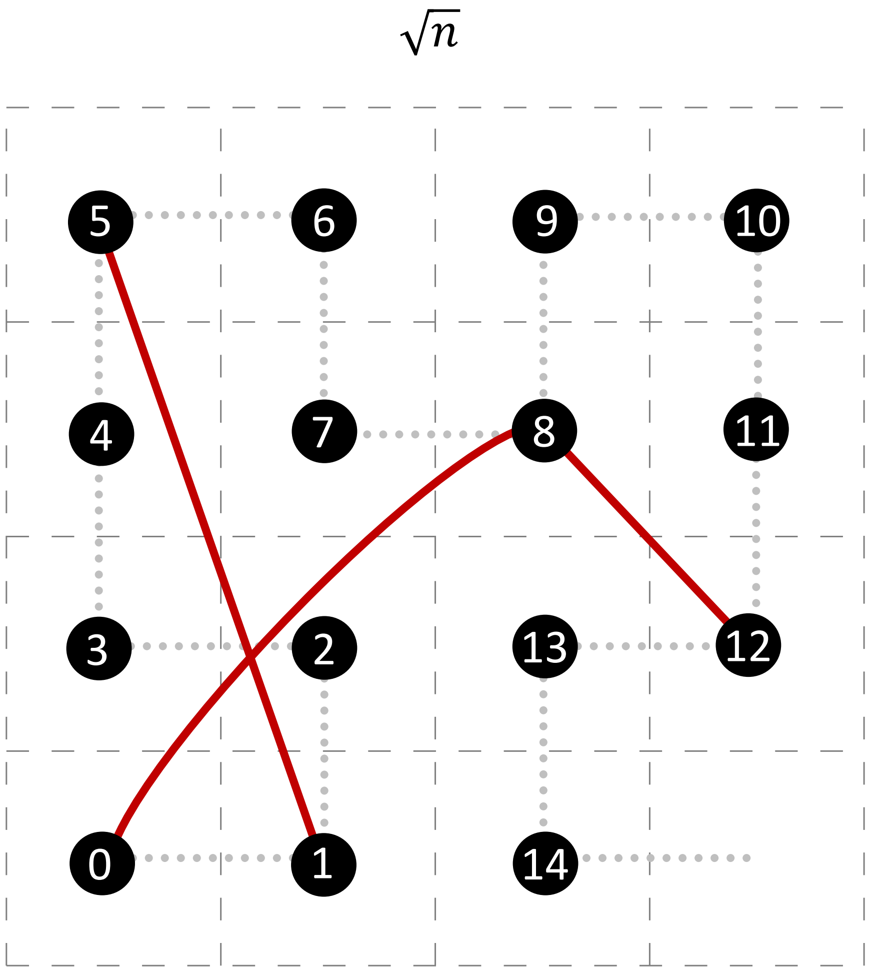
Message a neighbor in the ***tree****:*

1 hop: average distance $\Theta(\sqrt{n})$

Spatial Layout – Trees



DFS

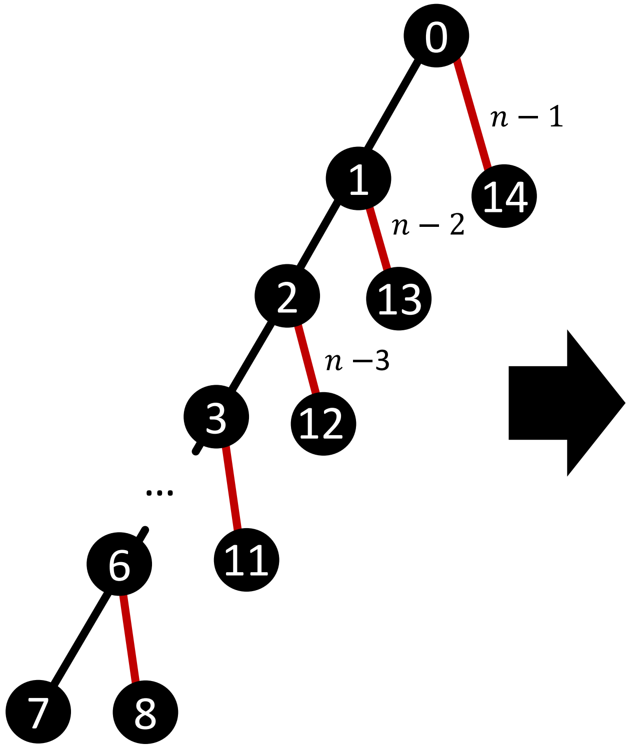


DFS+Hilbert

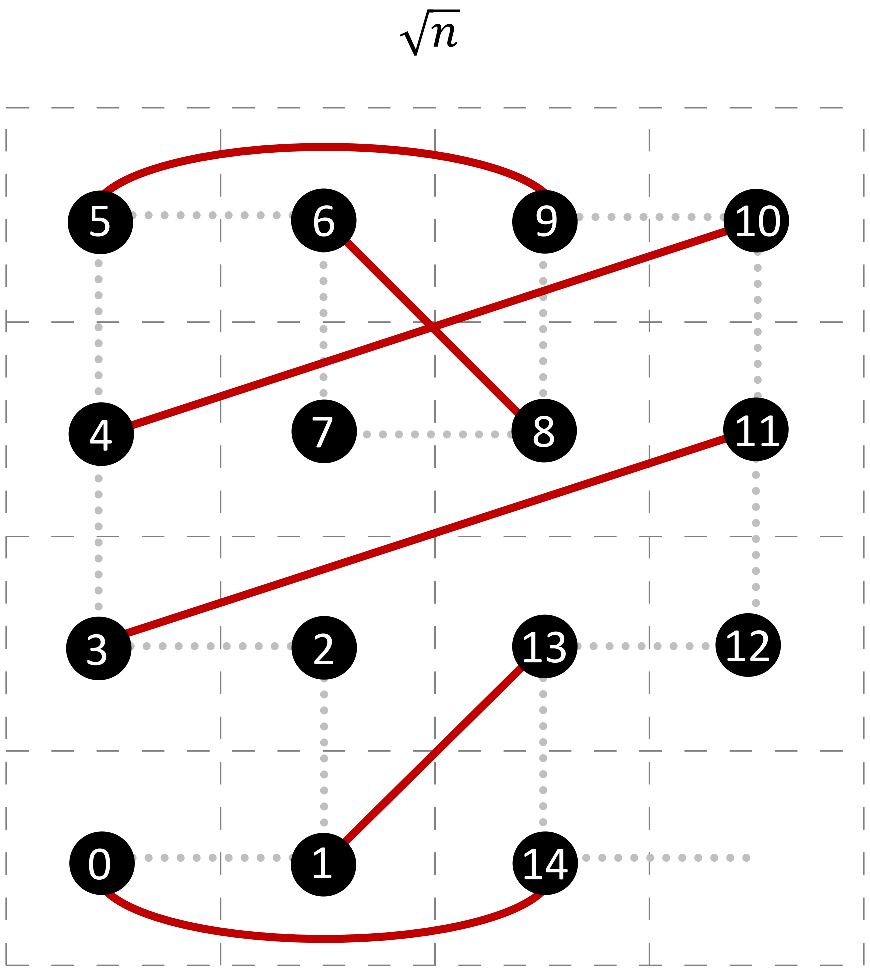
Message a neighbor in the ***tree****:*

1 hop: average distance $\Theta(1)$

Spatial Layout – Trees



DFS

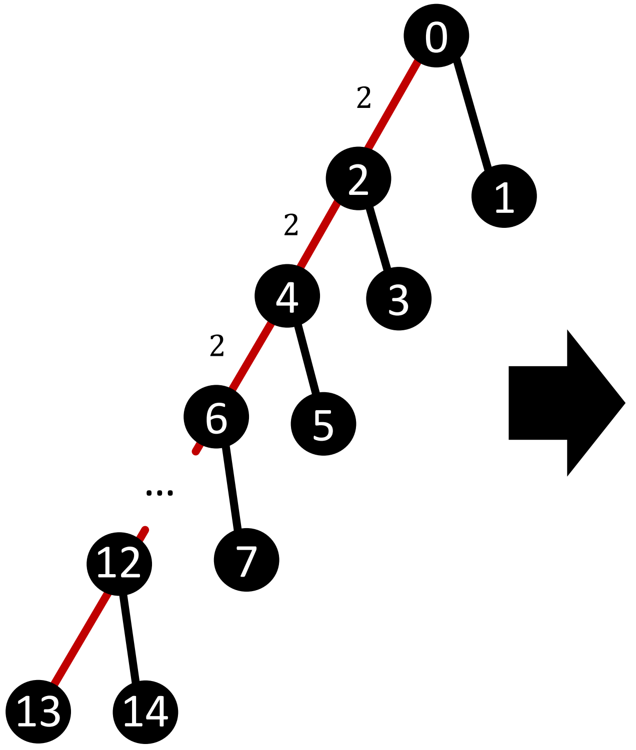


DFS+Hilbert

Message a neighbor in the ***tree****:*

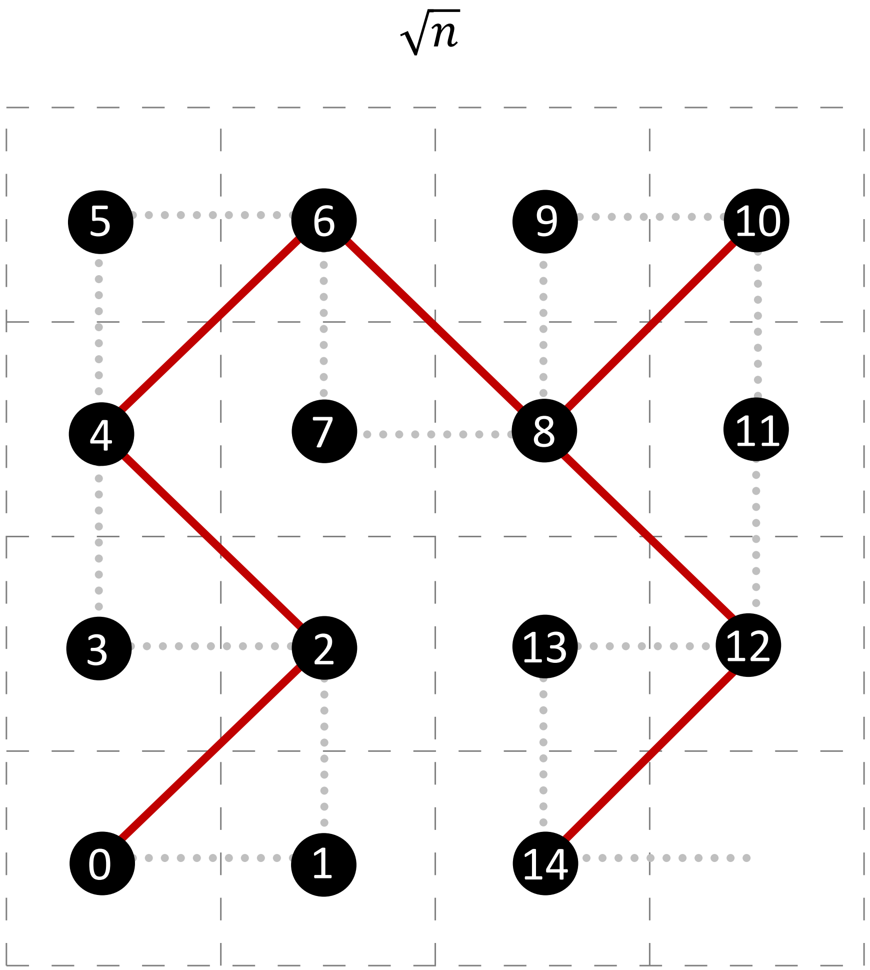
1 hop: average distance $\Theta(\sqrt{n})$

Spatial Layout – Trees



Recursively traverse children in order of increasing size

Light-First Order

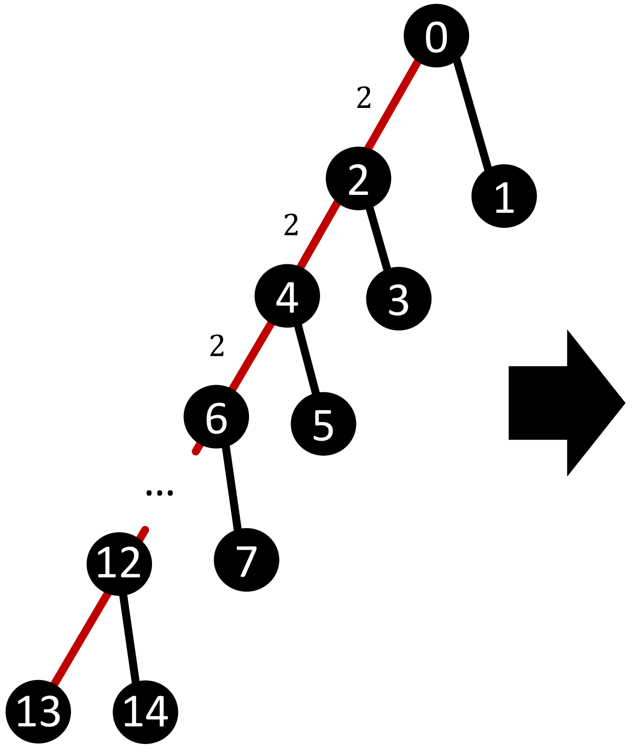


Light First+Hilbert

Message a neighbor in the ***tree****:*

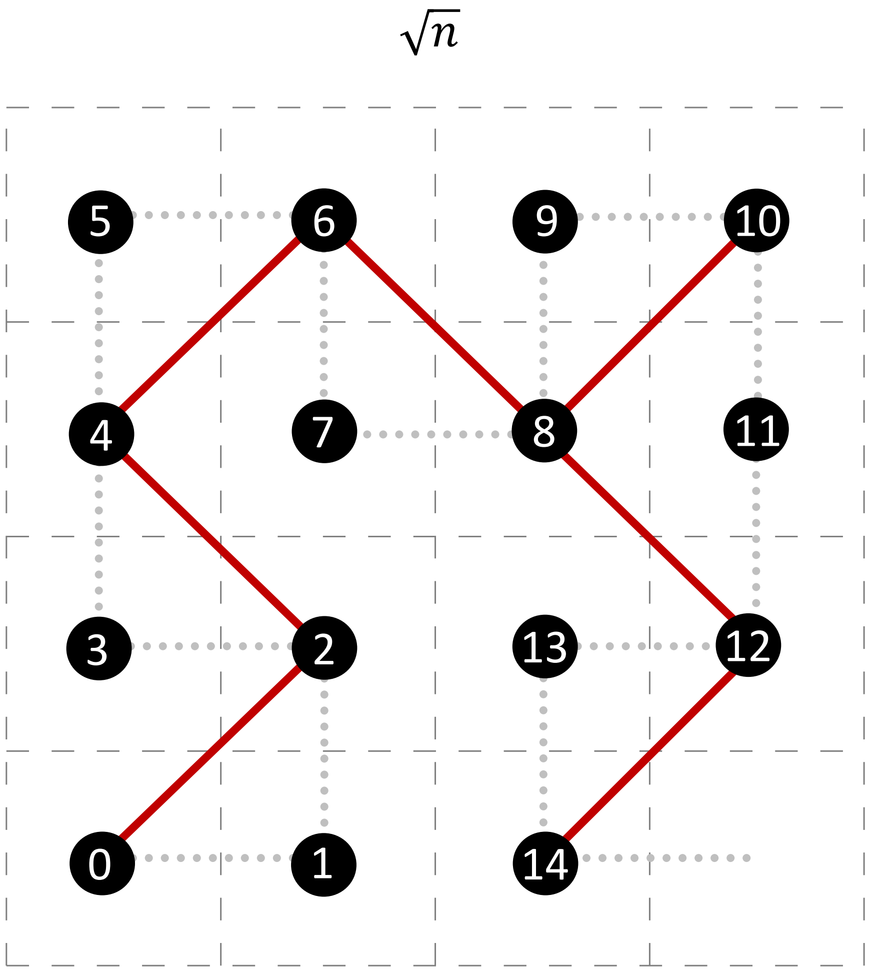
1 hop: distance ≤ 2

Spatial Layout – Trees



Recursively traverse children in order of increasing size

Light-First Order

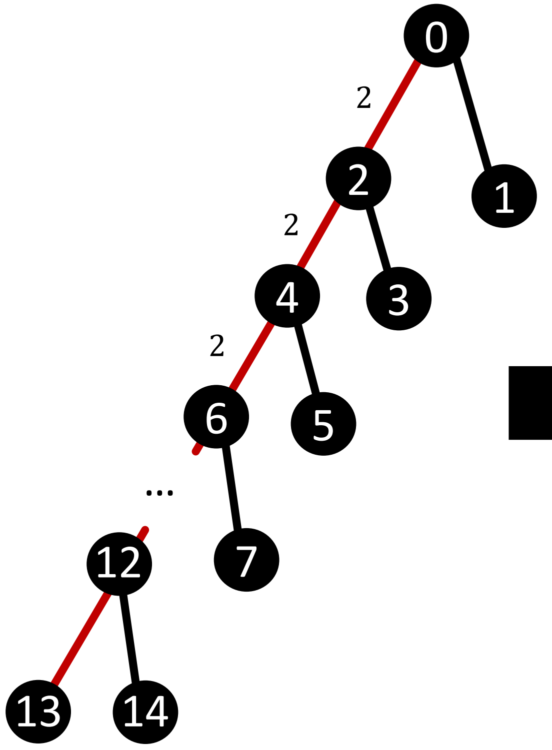


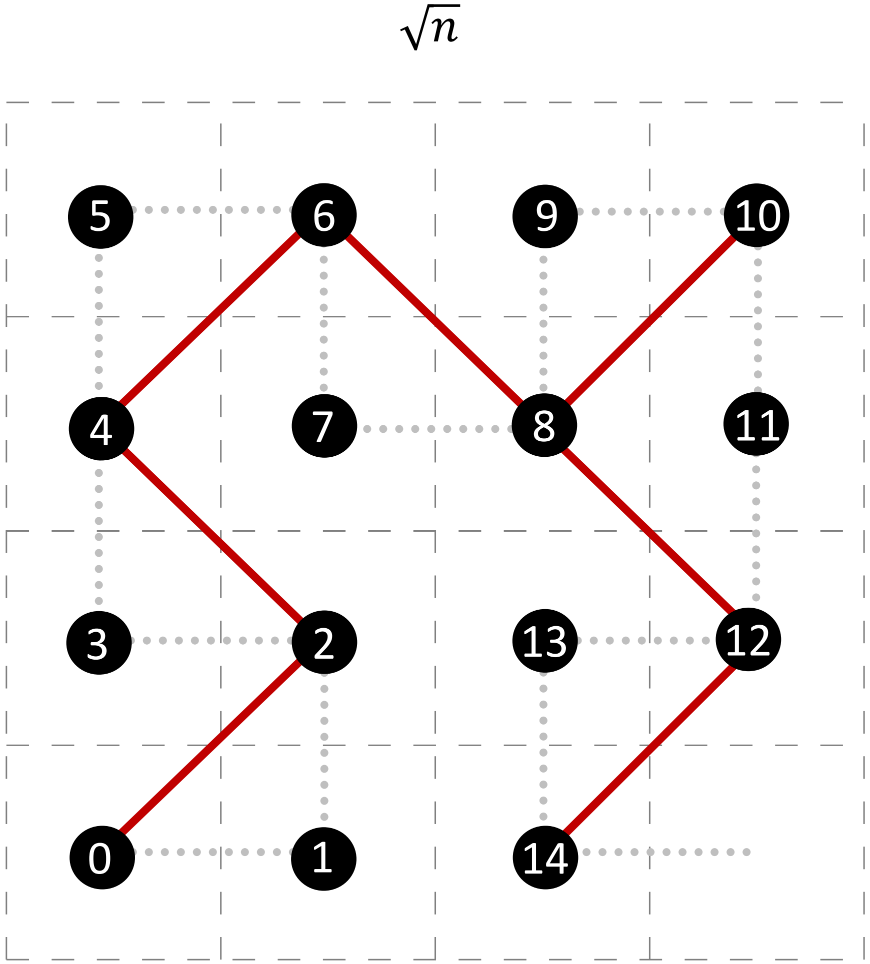
Light First+Hilbert

Message a neighbor in the ***tree****:*

For maximum degree d, Light-First + Hilbert guarantees **average distance $\mathbf{O(d)}$**

Spatial Layout – Trees





Recursively traverse children in order of increasing size

Light-First Order

Light First+Hilbert

Message a neighbor in the ***tree****:*

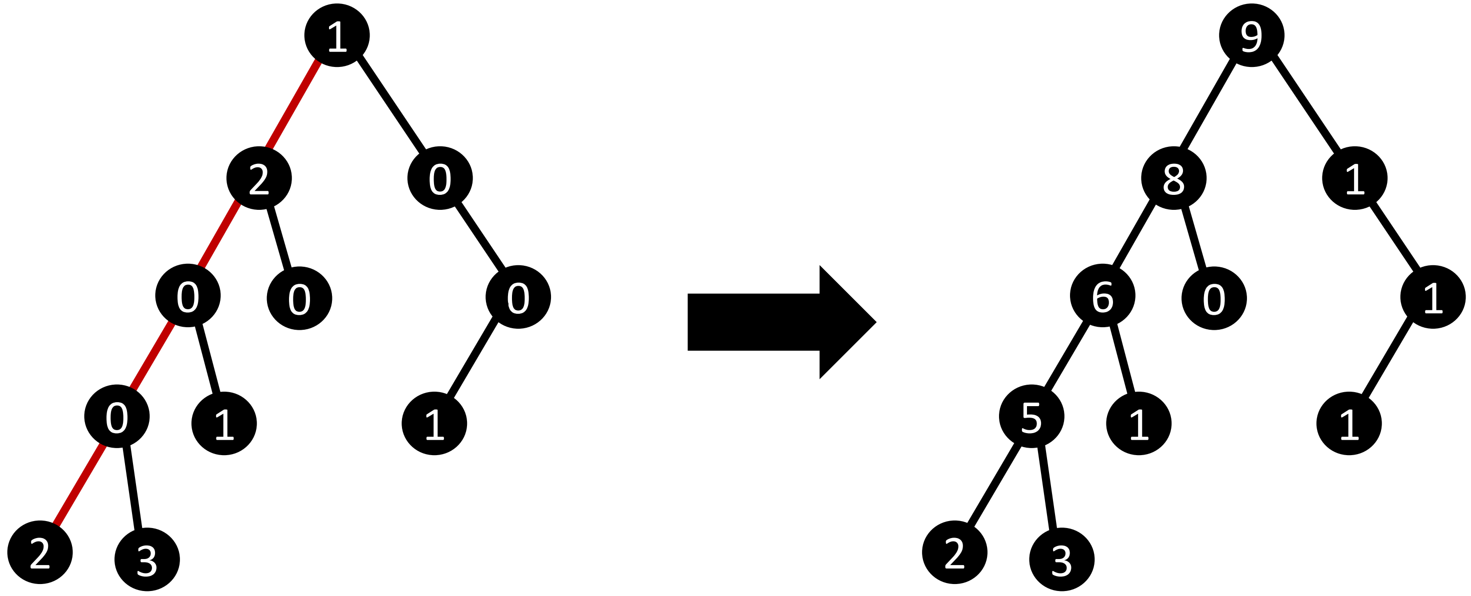
We can transform the tree into bounded degree

For maximum degree d, Light-First + Hilbert guarantees **average distance** $\mathbf{O(d)}$

Optimal up to constant factors

Other space-filling curves also work

Logical Operation – Treefix Sum



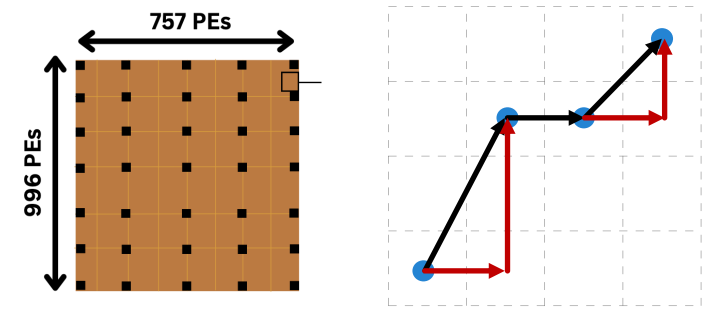
Idea:
Send partial sums to parents

Spatial layout guarantees
low average distance

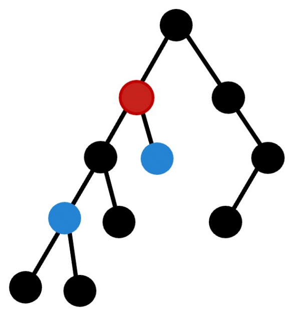
Naïve Approach:
$\Omega(n)$ ***depth***
n vertices

Prallel Tree Contraction:
$O(\log^2 n)$ ***depth*** *w.h.p.*

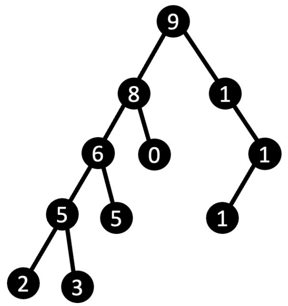
Conclusions



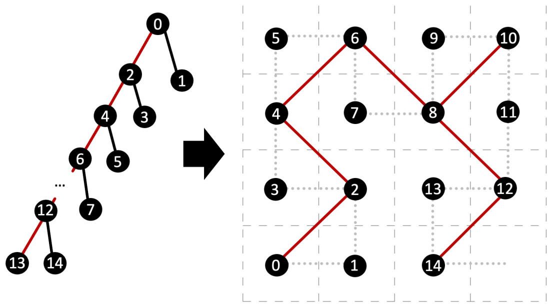
Spatial Model of Computation



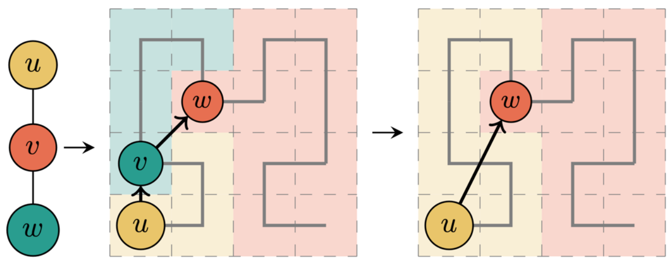
LCA



Treefix Sum



Near-Optimal Spatial Tree Layout



Efficient Tree Algorithms

More of SPCL's research:

- youtube.com/@spcl — 180+ Talks
- twitter.com/spcl_eth — 1.4K+ Followers
- github.com/spcl — 3.8K+ Stars

… or spcl.ethz.ch